CS 342 – Operating Systems

Project 01

Ümit Yiğit Başaran

21704103 – Section 1

28.02.2021

This report contains the outputs of the executions, graphs of the outputs, and a conclusion. The program were executed in the terminal with ‘./isp <N> <mode>’ command. The elapsed time are demonstrated in the output with green color. Number of ‘read’ and ‘write’ system calls, and character count are demonstrated in the output with blue color. Prompt string ‘isp$’ is demonstrated in the program with red color.

The M values 100, 1000, 10000, 100000, 1000000, and 10000000 were tested for each experiment where M is the number of random characters that the producer generates and the consumer requests. Because the buffer size (N value) is not important for the first mode (normal mode), there are no experiments where the N value was changed. **For the second mode (tapped mode), the N values 1, 4, 16, and 64** were tested, respectively. Also, because ‘read’ and ‘write’ system calls are not used in normal mode, the number of read/write operations and the number of characters is not counted in normal mode.

In the second mode, because read/write operations are done in a while loop and the condition of this loop is controlled by read system call, the read operation counter is incremented by one after exiting from while loop.

Outputs of The Experiments:

- **Output for normal mode:**

```
umityigitbsrn@ubuntu:~/Desktop/OS/projects/project01$ ./isp 1 1
isp$: ./producer 100 | ./consumer 100
Elapsed time: 0.001493
isp$: ./producer 1000 | ./consumer 1000
Elapsed time: 0.001817
isp$: ./producer 10000 | ./consumer 10000
Elapsed time: 0.003479
isp$: ./producer 100000 | ./consumer 100000
Elapsed time: 0.019597
isp$: ./producer 1000000 | ./consumer 1000000
Elapsed time: 0.157334
isp$: ./producer 10000000 | ./consumer 10000000
Elapsed time: 1.227323
isp$:
```

Figure 1: Normal mode

- **Output for tapped mode with buffer size = 1**

```
umityigitbsrn@ubuntu:~/Desktop/OS/projects/project01$ ./isp 1 2
isp$: ./producer 100 | ./consumer 100
character-count: 100
read-syscall-count: 101
write-syscall-count: 100
Elapsed time: 0.001271
isp$: ./producer 1000 | ./consumer 1000
character-count: 1000
read-syscall-count: 1001
write-syscall-count: 1000
Elapsed time: 0.003372
isp$: ./producer 10000 | ./consumer 10000
character-count: 10000
read-syscall-count: 10001
write-syscall-count: 10000
Elapsed time: 0.020539
isp$: ./producer 100000 | ./consumer 100000
character-count: 100000
read-syscall-count: 100001
write-syscall-count: 100000
Elapsed time: 0.405142
isp$: ./producer 1000000 | ./consumer 1000000
character-count: 1000000
read-syscall-count: 1000001
write-syscall-count: 1000000
Elapsed time: 1.637593
isp$: ./producer 10000000 | ./consumer 10000000
character-count: 10000000
read-syscall-count: 10000001
write-syscall-count: 10000000
Elapsed time: 14.626437
isp$:
```

Figure 2: Tapped mode – N = 1

- **Output for tapped mode with buffer size = 4**

```
isp$: ./producer 100 | ./consumer 100
character-count: 100
read-syscall-count: 26
write-syscall-count: 25
Elapsed time: 0.001750
isp$: ./producer 1000 | ./consumer 1000
character-count: 1000
read-syscall-count: 251
write-syscall-count: 250
Elapsed time: 0.002034
isp$: ./producer 10000 | ./consumer 10000
character-count: 10000
read-syscall-count: 2501
write-syscall-count: 2500
Elapsed time: 0.005877
isp$: ./producer 100000 | ./consumer 100000
character-count: 100000
read-syscall-count: 25001
write-syscall-count: 25000
Elapsed time: 0.051347
isp$: ./producer 1000000 | ./consumer 1000000
character-count: 1000000
read-syscall-count: 250001
write-syscall-count: 250000
Elapsed time: 0.919728
isp$: ./producer 10000000 | ./consumer 10000000
character-count: 10000000
read-syscall-count: 2500001
write-syscall-count: 2500000
Elapsed time: 13.522648
isp$:
```

Figure 3: Tapped mode – N = 4

- **Output for tapped mode with buffer size = 16**

```
umityigitbsrn@ubuntu:~/Desktop/OS/projects/project01$ ./isp 16 2
isp$: ./producer 100 | ./consumer 100
character-count: 100
read-syscall-count: 8
write-syscall-count: 7
Elapsed time: 0.001609
isp$: ./producer 1000 | ./consumer 1000
character-count: 1000
read-syscall-count: 64
write-syscall-count: 63
Elapsed time: 0.001927
isp$: ./producer 10000 | ./consumer 10000
character-count: 10000
read-syscall-count: 626
write-syscall-count: 625
Elapsed time: 0.003858
isp$: ./producer 100000 | ./consumer 100000
character-count: 100000
read-syscall-count: 6251
write-syscall-count: 6250
Elapsed time: 0.025018
isp$: ./producer 1000000 | ./consumer 1000000
character-count: 1000000
read-syscall-count: 62501
write-syscall-count: 62500
Elapsed time: 0.242619
isp$: ./producer 10000000 | ./consumer 10000000
character-count: 10000000
read-syscall-count: 625001
write-syscall-count: 625000
Elapsed time: 2.086201
isp$:
```

Figure 4: Tapped mode – N = 16

- **Output for tapped mode with buffer size = 64**

```
umityigitbsrn@ubuntu:~/Desktop/OS/projects/project01$ ./isp 64 2
isp$: ./producer 100 | ./consumer 100
character-count: 100
read-syscall-count: 3
write-syscall-count: 2
Elapsed time: 0.001614
isp$: ./producer 1000 | ./consumer 1000
character-count: 1000
read-syscall-count: 17
write-syscall-count: 16
Elapsed time: 0.001807
isp$: ./producer 10000 | ./consumer 10000
character-count: 10000
read-syscall-count: 158
write-syscall-count: 157
Elapsed time: 0.003557
isp$: ./producer 100000 | ./consumer 100000
character-count: 100000
read-syscall-count: 1564
write-syscall-count: 1563
Elapsed time: 0.021835
isp$: ./producer 1000000 | ./consumer 1000000
character-count: 1000000
read-syscall-count: 15626
write-syscall-count: 15625
Elapsed time: 0.153621
isp$: ./producer 10000000 | ./consumer 10000000
character-count: 10000000
read-syscall-count: 156251
write-syscall-count: 156250
Elapsed time: 1.505608
isp$:
```

Figure 5: Tapped mode – N = 64

Graphs:

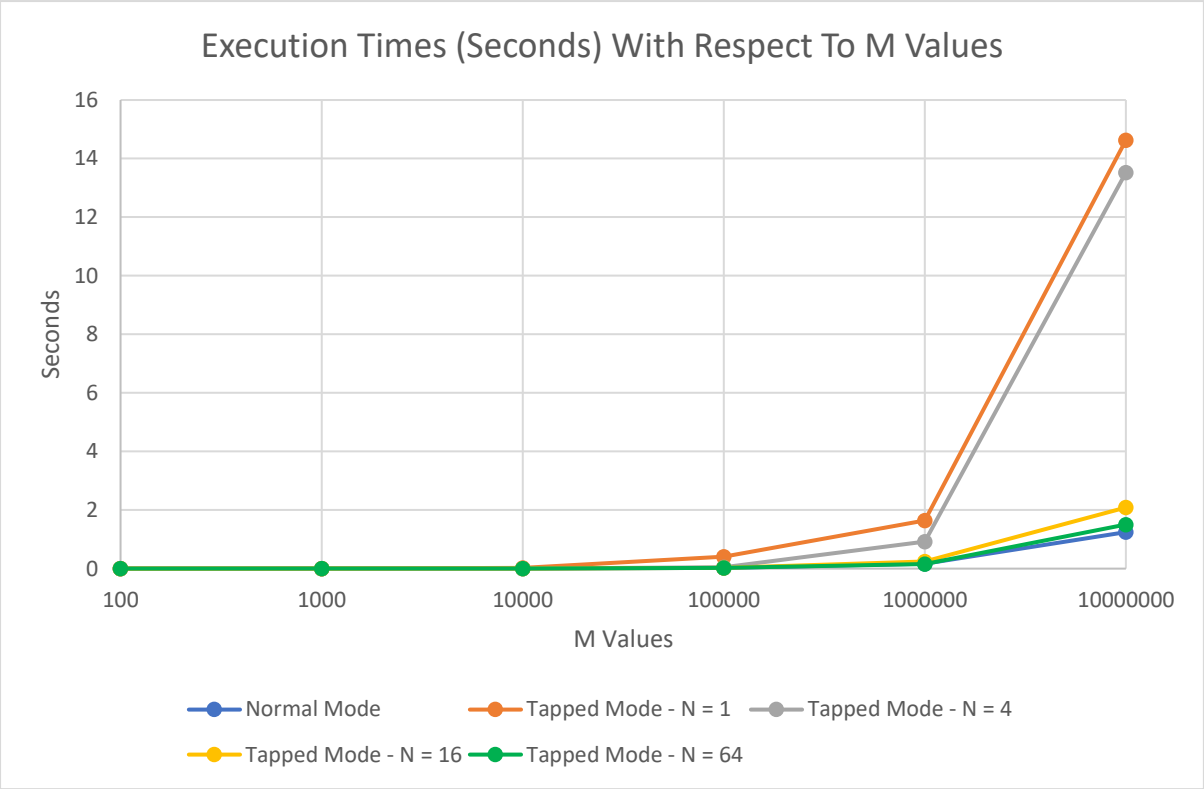

Figure 6: All experiments

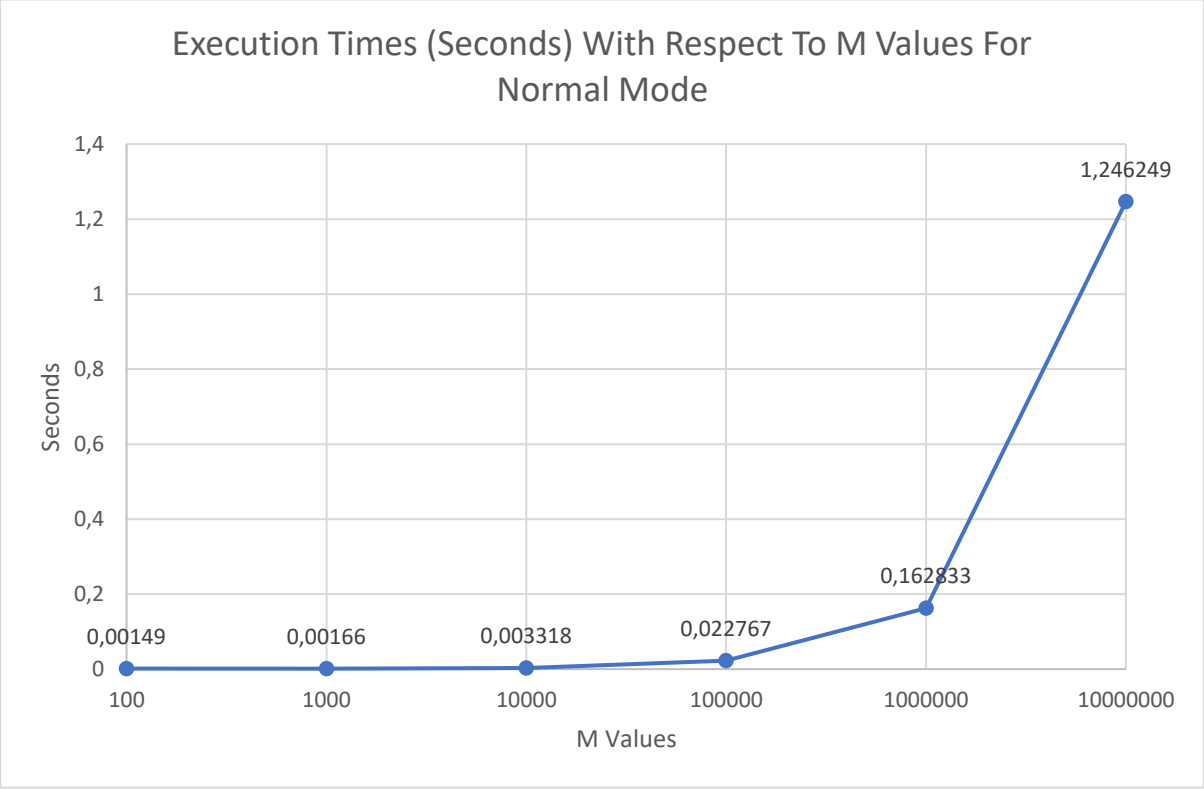

Figure 7: Experiment for normal mode

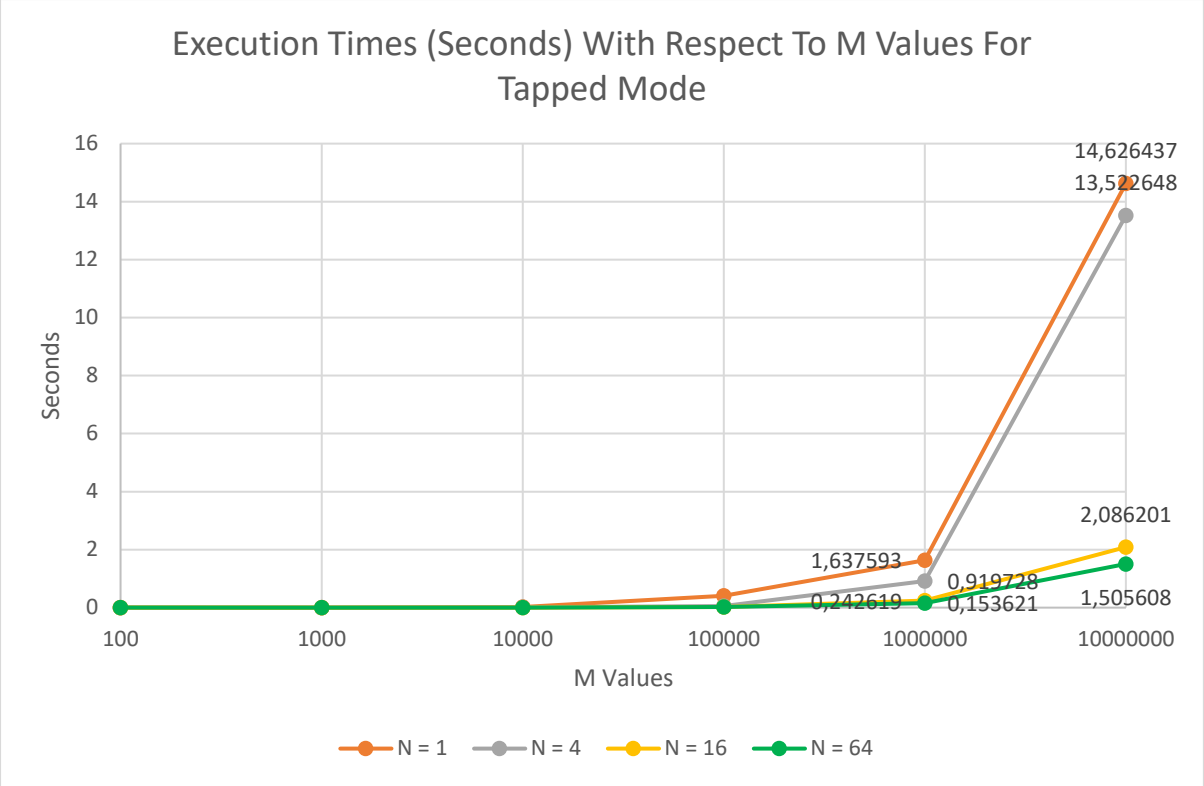

Figure 8: Experiments for tapped mode

Conclusion:

As a result, by looking at Figure 6, we can say that normal mode has the best execution time between the execution times of all experiments. Also, we can observe that as the buffer size (N) increases for the tapped mode, the result of the experiments gets closer and closer to the result of normal mode.

Because normal mode has no restriction while data transferring to one to child to another and there is a one pipe between the first and the second child, the normal mode takes a shorter amount of time. However, in tapped mode, there are two pipes (one connects the first child and the parent, other connects the parent and the second child) and there is a buffer which read data from the first pipe and writes it into the second pipe, so that, it makes more work than the normal mode and takes more amount of time. Therefore, when the buffer size is increased, the execution time for different M values is decreased. It can be observed more easily at higher M values (from Figure 7 and Figure 8).